User Manual for ReadSim V0.7

Ramona Schmid Daniel H. Huson

May 4, 2006

Contents

Contents 1

1 Introduction 2

2 Getting Started 3

3 Obtaining and Installing the Program 3

4 Program Overview 3

5 The Sanger Sequencing Model 4

6 The 454 Sequencing Model 4

7 Command-Line Options 6

8 Input Dialog 8

9 Console Window 8

10 File Formats 9

10.1 Input file 9

10.2 Output file 9

10.3 CGViz file 10

11 Examples 10

12 Acknowledgments 10

References 10

Index 12

1 Introduction

Disclaimer: This software is provided "AS IS" without warranty of any kind. This is developmental code, and we make no pretension as to it being bug-free and totally reliable. Use at your own risk. We will accept no liability for any damages incurred through the use of this software. Use of the ReadSim is free, however the program is not open source.

How to cite: If you publish results obtained in part by using ReadSim , then we require that you acknowledge this by citing the program as follows:

- R. Schmid, S.C. Schuster, M.A. Steel and D.H. Huson, *ReadSim- A simulator for Sanger and 454 sequencing*, in preparation, software freely available from www-ab.informatik.uni-tuebingen.de/software/readsim.

The predominant approach to sequencing large DNA molecules is *Sanger sequencing*, using a shotgun approach that involves cloning small pieces (or *inserts*) of DNA and then determining their sequence using fluorescent didesoxynucleotides for termination and capillary electrophoresis, see [4, 5]. In paired-end shotgun sequencing, both ends of an insert are read, see [8].

New sequencing technology based on pyro-sequencing, as recently introduced by 454 Life Sciences [2], promises to be ideally suited for metagenomic projects that seek to determine the diversity of organisms in a given environment. To develop and calibrate software to process the arising data, a sequencing simulator is required that samples reads from a given known target sequence under an error model that reflects the performance of the sequencing technology. Additionally, to simplify the comparison of pyro-sequencing and Sanger sequencing, it is desirable that both approaches can be simulated by a single program.

The aim of ReadSim is to simulate the process of collecting reads from a target genome. The current version of the program implements two different sequencing models.

The Sanger model simulates sequencing using Sanger sequencing, The simulation strategy and error model is modeled on the *celsim* program [6]. The 454 model simulates sequencing using the 454 approach, as described in [2]. The characteristics of this model differ substantially from the Sanger model, e.g. the collected reads are much shorter and mate-pairs are currently not available (but can be simulated using this software). The most important difference is that sequencing errors arise in a different way and usually involves the length of homopolymer runs.

ReadSim is a command-line program and is completely controlled by command line options. However, for convenience we also supply a console mode in which an `Input` dialog is presented and output is written to a `Console` window.

This document provides both an introduction and a reference manual for ReadSim .

2 Getting Started

This section describes how to get started.

First, download an installer for the program from

www-ab.informatik.uni-tuebingen.de/software/readsim,

see Section 3 for details. The executable program is called ReadSim . The unix and linux installers additionally provide an executable called ReadSim-cmdline.

Upon startup of ReadSim , a dialog is presented that can be used to set all command-line parameters. Press the `Help` button to list a summary of all available options. Use the `-i` option to specify an input file containing a target DNA sequence in FastA format. Use the `-x` option to specify the x-coverage desired. Use the option `-model` to choose between the 454 model and sanger model. By default, the program will use the 454 model and will write all collected reads to a file named `readsim.out`.

3 Obtaining and Installing the Program

ReadSim is written in Java and requires a Java runtime environment version 1.5 or newer, freely available from www.java.org.

ReadSim is installed using an installer program that is freely available from

www-ab.informatik.uni-tuebingen.de/software/readsim.

There are four different installers, targeting different operating systems:

- `ReadSim_windows_0_7.exe` provides an installer for Windows.
- `ReadSim_macos_0_7.sit` provides an installer for Mac OS.
- `ReadSim_linux_0_7.rpm` provides a RPM package for Linux.
- `ReadSim_unix_0_7.sh` provides a shell installer for Linux, MacOS and Unix.

  The executable program is called ReadSim . The unix and linux installers additionally provide an executable called ReadSim-cmdline.

4 Program Overview

In this section, we give an overview of the main design goals and features of this program. Basic knowledge of the underlying design of the program should make it easier to use the program.

ReadSim is written in the programming language Java. The advantages of this is that we can provide versions that run under the Linux, MacOS, Windows and Unix operating systems. A potential draw-back is that an algorithm implemented in Java will generally run slower than the same algorithm implemented in C or $C++$.

The program is designed to be run from the command-line, using switches such as `-i genome.fasta` to set all aspects of a simulation. However, for the convenience of use on operating systems that do not provide easy access to the command-line (such as Windows), by default, the program provides an `Input` dialog to enter the configuration options and a `Console` window to report the progress of the program.

5 The Sanger Sequencing Model

Most major sequencing projects are based on the same experimental technique, called *shotgun sequencing*, based on the method originally developed by Fred Sanger and his group. We will refer to this as the *Sanger* sequencing model.

This technique is based on automated gel sequencers that use electrophoresis and fluorescent markers to determine the sequence of the nucleotides. The ability of these machines to read consecutive pieces of DNA degrades quickly with the length of the sequence, and today a sequencing machine can read up to ≈ 1000 consecutive base pairs of a fragment of DNA, depending on the degree of accuracy desired.

The sequence of a fragment determined in this way is called a *read*. The fragments are sampled from a stretch of DNA that is often referred to as the *source sequence* (the sequence that we take the fragment from) or as the *target sequence* (the sequence we want to reconstruct from the reads).

To model Sanger sequencing, the ReadSim program is supplied with a target genome sequence. Fragments are then sampled from the target sequence. To simulate the sequencing process, the length of reads is randomly chosen under a uniform or normal distribution and sequencing errors are introduced into the reads. The frequency of sequencing errors is position dependent, following a ramp distribution.

Additionally, to simulate paired-end sequencing, the length of inserts is randomly chosen under a uniform or normal distribution, and failed pairs and chimeric pairs are produced.

Typical values for Sanger sequencing is a mean read length of 700 bp and 7-fold coverage, and insert sizes of 2kb, 10kb, 50kb or 150kb.

The options provided for this sequencing model follow those provided by the program `celsim` [6].

6 The 454 Sequencing Model

Here we describe the *454* sequencing model.

Recently, *454 Life Sciences* announced a new, highly parallel sequencing system with significantly higher throughput than achievable with previous methods, see [3]. It uses emulsion-based PCR amplification of a large number of DNA fragments and high-throughput parallel pyro-sequencing. The system is reportedly able to sequence 25 million bases within four hours. Cloning of the target DNA fragments is not necessary and the method is much cheaper, per base, than other existing methods. Drawbacks of the method are shorter read lengths of about 100 bases, in contrast to 800 bases using Sanger sequencing, and a higher error rate. Further, sequencing of paired-end reads is not yet possible.

The *454* sequencing approach promises to be very suitable for metagenomics projects, as demonstrated by [7]. To develop and calibrate software to analyze the data arising in this context, we have developed a program `ReadSim` that simulates the sequencing process by generating a collection of reads from a given *target* genome and introducing sequencing errors into the reads, based on error models reflecting the properties of the sequencing technology. Our program simulates both Sanger sequencing and 454 sequencing, and in both cases can simulator pair-end sequencing, although the latter is not yet possible with 454 technology.

In pyro-sequencing, the intensity of emitted light is used to estimate the length of homopolymers, i.e. runs of identical nucleotides in a sequence. During sequencing, the four DNA composing nucleotides are periodically *flowed* over the inserts to be sequenced. Within each flow, the intensity of the signal emitted reflects the number of nucleotides incorporated and thus the length of the homopolymer under consideration. For chemical and technical reasons, this signal is subject to fluctuations that lead to sequencing errors. [3] report an error rate of about 3%.

The `ReadSim` simulator produces reads with normally or uniformly distributed lengths. They are randomly sampled from the target sequence and then subjected to a simulation of sequencing errors, as follows.

Let r denote the length of a given homopolymer. We model the emitted light intensity using a normal distribution $N(\mu, \sigma)$, with mean $\mu = r$ and standard deviation $\sigma = k \cdot \sqrt{r}$, where k is a fixed proportionality factor. Following [3], by default we use $k = 0.15$. Although basic statistics implies that the standard deviation should grow with the square root of r, in [3] the standard deviation of the light intensity emitted during 454-sequencing is reported to be $\sigma = k \cdot r$. Both variants of the calculation are implemented in our software.

A *negative flow* is a flow of nucleotides in which the sequence it not elongated. Light intensities of negative flows follow a log-normal distribution, with mean $\mu = 0.23$, and standard deviation $\sigma = 0.15$, see [3]. A random variable X is lognormally distributed if and only if the random variable $\ln(X)$ is normally distributed. Let μ and σ be the mean and standard deviation of X and let m and s be the mean and standard deviation of $\ln(X)$. Then:

$$m = \ln\left(\frac{\mu^2}{\sqrt{\sigma^2 + \mu^2}}\right) \quad \text{and} \quad s = \sqrt{(\ln\left(\left(\frac{\sigma}{\mu}\right)^2 + 1\right).}$$

The probability density function of a lognormal distributed random variable is usually specified with mean m and standard deviation s of the underlying normally distributed $\ln(X)$, as stated in the following:

$$f(x) = \begin{cases} \frac{e^{-\frac{1}{2}\left(\frac{\ln(x)-m}{s}\right)^2}}{xs\sqrt{2\pi}}, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Based on this, we simulate base-calling intensities of negative flows and model the miss-interpretation of null-mers as homopolymers of length = 1. Our algorithm takes the order of the nucleotide flows into account and so after a given base only two specific negative flows in a specific order are allowed, since the nucleotides are cyclicly flowed in the order T,A,C,G.

We have implemented two different methods of base-calling. In the *intersection* model, the intersections of the density functions $f_{N(r_1,\sigma)}(x)$, and $f_{N(r_2,\sigma)}(x)$ of the normal distributions for different homopolymer lengths r_1 and r_2 are calculated and stored in an *intersection matrix* M. These values are used to decide which homopolymer length is called. Each entry of the matrix holds the x-value of the intersection of two normal distributions, i.e. $M(r, r-1)$ holds the intersection of $f_{N(r,\sigma)}(x)$, and $f_{N(r-1,\sigma)}(x)$, where σ is set according to the chosen intensity calculation.

The *intersection-weight* method extends *intersection* base-calling by using an additional binary experiment with probabilities weighted by $f_{N(r,\sigma)}(i)$ and $f_{N(r\pm1,\sigma))}(i)$, where i is the intensity of light generated according to the chosen $N(r,\sigma)$-distribution.

7 Command-Line Options

The ReadSim program is controlled by options that are either provided on the command-line, or using the `Options` dialog. Here we list all available options. Use the `-h` option to obtain a listing of all options directly from the program.

Main options:

`-i` *name*	Name of input file containing source genome.
`-c`	When set, source genome is treated as circular.
`-o` *name*	Name of output file. Default is `readsim.out`.
`-or`	When set, existing output file is replaced.
`-model` *model*	Sequencing model, either `454` or `sanger`.

Options for read sequencing:

`-x`	*number*	Desired sequencing x-coverage. If set, overrides option `-n`.
`-n`	*number*	Number of reads to generate. Default is 2000.
`-modlr`	*model*	Read-length model, either `uniform` or `normal`. Default is: `uniform`.
`-minlr`	*number*	Minimal length of read (uniform model). Default is 80.
`-maxlr`	*number*	Maximal length of read (uniform model). Default is 120.
`-meanlr`	*number*	Mean length of reads to generate (normal model). Default is 100.
`-stdlr`	*number*	Standard deviation as proportion of mean (normal model). Default is 0.1.
`-f`	*number*	Probability that read is chosen from forward strand. Default is 0.5.

Options for insert sequencing:

`-modli`	*model*	Insert length model, either `uniform` or `normal`. Default is `uniform`.
`-minli`	*number*	Minimal length of insert (uniform model). Default is 1600.
`-maxli`	*number*	Maximal length of insert (uniform model). Default is 2400.
`-meanli`	*number*	Mean length of insert lengths (normal model). Default is 2000.
`-stdli`	*number*	Standard deviation as proportion of mean (normal model). Default is 0.1.
`-pf`	*number*	Proportion of reads that won't be paired. Default is 0.1.
`-pc`	*number*	Proportion of chimeric inserts. Default is 0.05.

Options for the Sanger sequencing model:

`-eb`	*number*	Single base error rate at start of a read. Default is 0.01.
`-ee`	*number*	Single base error rate at end of a read. Default is 0.02.
`-pd`	*number*	Proportion of errors that are deletions. Default is 0.2.
`-pi`	*number*	Proportion of errors that are insertions. Default is 0.2.

Options for the `454` sequencing model:

`-sqrt`		If set, use $\sigma = \langle -k \rangle \times \sqrt{(length)}$, else use $\sigma = \langle -k \rangle \times (length)$. Default is "set".
`-k`	*number*	Proportionality factor for calculating standard deviation proportional to sqrt(homopolymer length). Default is 0.15.
`-bc`	*model*	Base-calling options. `inter`: calculate intersection of normal density curves, `interw`: intersection+weight, and `nearest`: rounds to nearest integer. Default is: `inter`.
`-454mate`		If set, enable generation of mate pairs for 454 model, else disabel.

Additional options:

`-pre` *label*	Prefix for read identifiers. Default is `read`.
`-start` *number*	Number of first read identifier. Default is 1.
`-cgv` *name*	Save CGViz graphics to this file, if provided.
`-seed` *number*	If non-zero, use as seed for random number generator.
`-v`	Verbose: show settings of command-line arguments.
`-V`	Show version information.
`-h`	Show usage and quit.

8 Input Dialog

The command-line version ReadSim-cmdline is configured by specifying options on the command line when launching the program. At startup, the other version ReadSim opens an `Options` dialog which must be used to configure the program. The Options dialog is modal and will only close when either the `Help`, `Cancel` or `Apply` button is pressed.

Options are typed into the *input-area* in the same syntax as used for command-line options, as described in Section 7. Options typed into this area can be saved to a file using the `Save Options` button, or read from a file using the `Read Options` button.

Pressing the `Help` button will close the dialog and request the program to provide a list of all options in the console window. Pressing the `Cancel` button will close the dialog without setting any options. Pressing the `Apply` button run the program using all options specified in the input-area. As the program is currently not re-entrant, after pressing any of these three buttons requires the the program is closed and then re-launched to get back to the option dialog.

9 Console Window

The command-line version ReadSim-cmdline writes all messages to the shell in which it was launched. The other version ReadSim opens a `Console` window and writes all messages to it. This window has the usual text-related menu items.

The `File` menu contains the following file-related items:

- The `File→Save As` item saves the contents of the console window to a file. The `File→Print` item prints the contents of the console window.
- The `File→Close` item quits the program.

The `Edit` menu contains the usual edit-related items:

- The `Edit→Cut` item is used to cut text in the console window.

- The `Edit→Copy` item is used to copy text.
- The `Edit→Paste` item is used to paste text.
- The `Edit→Select All` item is used to select the whole text.
- The `Edit→Clear` item is used to clear the console window.

10 File Formats

The program reads and writes sequence data in FastA format.

10.1 Input file

ReadSim requires that the input file contains a single FastA record of DNA sequence.

10.2 Output file

The output is written in FastA format, one record per read. The header line of each record is structured as follows:

```
><read id> |beg|<start>|length|<length>|<forward|reverse>|<original FastA header>
```

or alternatively if a mate pair exists:

```
><read id> |beg|<start>|length|<length>|<forward|reverse>|matePair|...
...<read id mate>|chimeric|<true|false>|<original FastA header>
```

There, <read id> obviously is the id generated for the read belongig to that entry, <start> states the position within the source sequence, the read is taken from, <length> states the length of the produced read, <forward|reverse> tells you whether the read is taken form the forward or the reverse strand, <read id mate> holds the id of the mate pair if applicable, <true|false> is set to true if the mate pair is a chimeric one, else to false. <original FastA header> holds the origial FastaA header of the input sequence. The sequence of the read is stated in the following lines.

10.3 CGViz file

Optionally, the simulator can write information in CGViz format, using the `-cgv` option. This file can be opened using the *CGViz* program [1] and provides an interactive visualization of the sampled reads. The CGViz program is obtainable from: www-ab.informatik.uni-tuebingen.de/software/cgviz.

11 Examples

Example files are provided with the program. They are contained in the `examples` sub directory of the installation directory. The precise location of the installation directory depends upon your operating system.

The file `human-mtdna.fasta` contains the complete sequence of a human mtDNA genome. The file `human-mtdna-454.reads` contains a collection of 10-x coverage of 454 reads. The file `human-mtdna-sanger.reads` contains a collection of 10-x coverage of Sanger reads from inserts.

12 Acknowledgments

We would like to thank Gene Myers, Jonathan Rothberg, Lei Du and Erick Matsen for helpful discussions.

References

[1] O. Delgado Friedrichs, T. Dezulian, and D. H. Huson. A meta-viewer for biomolecular data. *GI Jahrestagung*, 1:375–380, 2003.

[2] M. Margulies, M. Egholm, W.E. Altman, S. Attiya, J.S. Bader, L.A. Bemben, J. Berka, M.S. Braverman, Y.-J. Chen, Z. Chen, S.B. Dewell, L.D., J.M. Fierro, X.V. Gomes, B.C. Godwin, W. He, S. Helgesen, C.H. Ho, G.P. Irzyk, S.C. Jando, M.L.I. Alenquer, T.P. Jarvie, K.B. Jirage, J.-B. Kim, J.R. Knight, J.R. Lanza, J.H. Leamon, S.M. Lefkowitz, M. Lei, J. Li, K.L. Lohman, H. Lu, V.B. Makhijani, K.E. McDade, M.P. McKenna, E.W. Myers, E. Nickerson, J.R. Nobile, R. Plant, B.P. Puc, M.T. Ronan, G.T. Roth, G.J. Sarkis, J.F. Simons, J.W. Simpson, M. Srinivasan, K.R. Tartaro, A. Tomasz, K.A. Vogt, G.A. Volkmer, S.H. Wang, Y. Wang, M.P. Weiner, P. Yu, R.F. Begley, and J.M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[3] M. Margulies and *et al.* Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[4] D. Meldrum. Automation for genomics, part one: Preparation for sequencing. *Genome Research*, 10(8):1081–1092, 2000.

[5] D. Meldrum. Automation for genomics, part two: Sequencers, microarrays, and future trends. *Genome Research*, 10(9):1288–1303, 2000.

[6] G. Myers. A dataset generator for whole genome shotgun sequencing. In *Proc Int Conf Intell Syst Mol Biol*, pages 202–10, 1999.

[7] H. N. Poinar, C. Schwarz, Ji Qi, B. Shapiro, R. D. E. MacPhee, B. Buigues, A. Tikhonov, D.H. Huson, L. P. Tomsho, A. Auch, M. Rampp, W. Miller, and S. C. Schuster. Metagenomics to Paleogenomics: Large-Scale Sequencing of Mammoth DNA. *Science*, 331:392–394, 2006.

[8] J. L. Webber and E. W. Myers. Human whole-genome shotgun sequencing. *Genome Research*, 7(5):401–409, 1997.

Index

-454mate, 7
-V, 8
-bc, 7
-c, 6
-cgv, 8
-eb, 7
-ee, 7
-f, 7
-h, 8
-i, 6
-k, 7
-maxli, 7
-maxlr, 7
-meanli, 7
-meanlr, 7
-minli, 7
-minlr, 7
-model, 6
-modli, 7
-modlr, 7
-n, 7
-o, 6
-or, 6
-pc, 7
-pd, 7
-pf, 7
-pi, 7
-pre, 8
-seed, 8
-sqrt, 7
-start, 8
-stdli, 7
-stdlr, 7
-v, 8
-x, 7

454, 4

Apply, 8

Cancel, 8
CGViz, 10
Clear, 9
Close, 8
Console, 8
Copy, 9
Cut, 8

Disclaimer, 2

Edit, 8
Edit→Clear, 9
Edit→Copy, 9
Edit→Cut, 8
Edit→Paste, 9
Edit→Select All, 9
examples, 10

File, 8
File→Close, 8
File→Print, 8
File→Save As, 8

Help, 8
How to cite, 2
human-mtdna-454.reads, 10
human-mtdna-sanger.reads, 10
human-mtdna.fasta, 10

i, 7
input-area, 8

Linux, 3, 4

Mac OS, 3
MacOS, 3, 4

Options, 8

Paste, 9
Print, 8

R, 3, 8
read, 4
Read Options, 8
ReadSim_linux_0_7.rpm, 3
ReadSim_macos_0_7.sit, 3
ReadSim_unix_0_7.sh, 3
ReadSim_windows_0_7.exe, 3

Sanger, 4
Save As, 8

Save Options, 8
Select All, 9
shotgun sequencing, 4
source sequence, 4

target sequence, 4

Unix, 3, 4

Windows, 3, 4